

APLICAÇÃO DE UM PROCESSO DE AQUISIÇÃO DE CONHECIMENTO NA PRÉ-MODELAGEM DE SOFTWARE

APPLICATION OF A KNOWLEDGE ACQUISITION PROCESS IN SOFTWARE PRE-MODELING

Lucas de Jesus Lima*
Franciene Duarte Gomes**

RESUMO

O objetivo deste artigo foi aplicar um processo de aquisição e representação do conhecimento intitulado de Verbka na pré-modelagem de software, para apoiar a construção do diagrama de classe em nível conceitual. Para isso foi utilizado a descrição de um sistema de uma empresa real e elaborado dois diagramas de classe. Um diagrama foi elaborado de modo tradicional, ou seja, sem o uso de um processo ou técnica de apoio. Para a elaboração do segundo diagrama utilizou-se os resultados da aplicação do processo Verbka com base na descrição do sistema. O principal resultado mostrou que o uso do Verbka na pré-modelagem pode apontar as principais classes de um projeto de software, e contribuir para a realização de inferências e identificação de novos processos de negócio. Conclui-se que o uso do processo Verbka na pré-modelagem pode colaborar para identificação de requisitos de software.

Palavras-chave: Nível conceitual. Extração de Conhecimento. Diagrama de Classes.

ABSTRACT

The objective of this article was to apply a process of acquisition and representation of the knowledge called Verbka in the pre-modeling of software, to support the construction of the class diagram at a conceptual level. For this, the description of a system of a real company was used and two class diagrams were elaborated. A diagram was drawn up in the traditional way, that is, without the use of a support process or technique. For the elaboration of the second diagram, the results of the application of the Verbka process were used based on the description of the system. The main result showed that the use of Verbka in pre-modeling can point out the main classes of a software project, and contribute to making inferences and identifying new business processes. It is concluded that the use of the Verbka process in pre-modeling can collaborate to identify software requirements.

Keywords: Conceptual level. Knowledge Extraction. Class Diagram.

* Universidade Federal de Mato Grosso do Sul Campus de Três Lagoas Mato Grosso do Sul.
lucasdjl.djl@gmail.com

** Universidade Federal de Mato Grosso do Sul Campus de Três Lagoas Mato Grosso do Sul.
franciene.gomes@ufms.br

Introdução

A engenharia de requisitos é o processo de compreensão e definição dos serviços requisitados do sistema e identificação de restrições relativas à operação e ao desenvolvimento do sistema (SOMMERVILLE, 2019). Assim, o processo da engenharia de requisitos tem como objetivo produzir um documento de requisitos acordados que especifica um sistema que satisfaz os requisitos dos *stakeholders* (SOMMERVILLE, 2019). Os *stakeholders* incluem os usuários finais que irão interagir com o sistema e qualquer outra pessoa em uma organização.

Para que seja possível essa interação, uma solução para o cliente é a aplicação de métodos com objetivo de melhoria da comunicação e do processo de desenvolvimento de novas soluções de tecnologias, para diferentes tipos de usuários. Sommerville (2019) aponta que o processo de levantamento de requisitos pode envolver um ou mais modelos de sistemas e protótipos. Com isso, podemos compreender de uma melhor maneira o sistema especificado, desde a construção de sua ideia até o seu desenvolvimento.

Durante a construção da ideia, alterações são realizadas desde o início do projeto, até o seu término. Frequentemente, durante o processo de levantamento de requisitos, é necessário modificar as entidades e relacionamentos criados para atender a uma necessidade dos usuários. Estas alterações são feitas fundamentadas em experiência, e muitas vezes podem fornecer ao usuário um sistema que não atende suas necessidades. Dessa maneira, torna-se necessário constantes mudanças gerando desgastes durante o ciclo de vida do sistema. Com base em tais informações, e tendo em vista a necessidade de uma estratégia de levantamento de requisitos mais eficiente é proposto neste artigo a aplicação do processo Verbka (VASQUES *et al.*, 2016) na pré-modelagem de software.

O presente trabalho busca responder a seguinte questão de pesquisa: Como a aplicação de um processo de aquisição de conhecimento, aplicado na pré-modelagem de software, pode apoiar a construção do diagrama de classe em nível conceitual? Sendo assim, o objetivo foi aplicar o processo Verbka (VASQUES *et al.*, 2016) na pré-modelagem de software, para apoiar a construção do diagrama de classe em nível conceitual.

A partir deste ponto este artigo está dividido do seguinte modo: na Seção 2 é apresentada a metodologia utilizada; a Seção 3 discute a fundamentação teórica que introduz os principais conceitos abordados, além de alguns trabalhos relacionados; a Seção 4 apresenta o processo Verbka; na Seção 5 apresenta a aplicação do processo e na

Seção 6 apresenta-se a análise dos resultados, e por fim na Seção 7 são apresentadas as conclusões e trabalhos futuros.

1 Metodologia

O trabalho de pesquisa segue uma abordagem qualitativa. De acordo com Creswell e Clark (2015), uma pesquisa qualitativa procura trazer a perspectivas de indivíduos envolvidos com o processo mostrando seu ponto de vista. Também foi realizado uma pesquisa bibliográfica para identificar trabalhos cujo foco foi na extração de conhecimento aplicados na modelagem de software.

Para atender ao objetivo do trabalho, foram elaborados dois diagramas de classe com base em uma descrição da visão de negócio de um caso real. Para a elaboração do primeiro diagrama de classes foi utilizado o processo de aquisição e representação do conhecimento Verbka (VASQUES *et al.*, 2016) para extração dos conceitos e seus relacionamentos de acordo com descrição da visão de negócio. Para a elaboração do segundo diagrama de classe foi utilizado os conceitos que compõem a construção de diagrama de classe da UML (BOOCH; RUMBAUGH; JACOBSON, 2012). A Figura 1 ilustra o sequenciamento das atividades feito neste estudo.

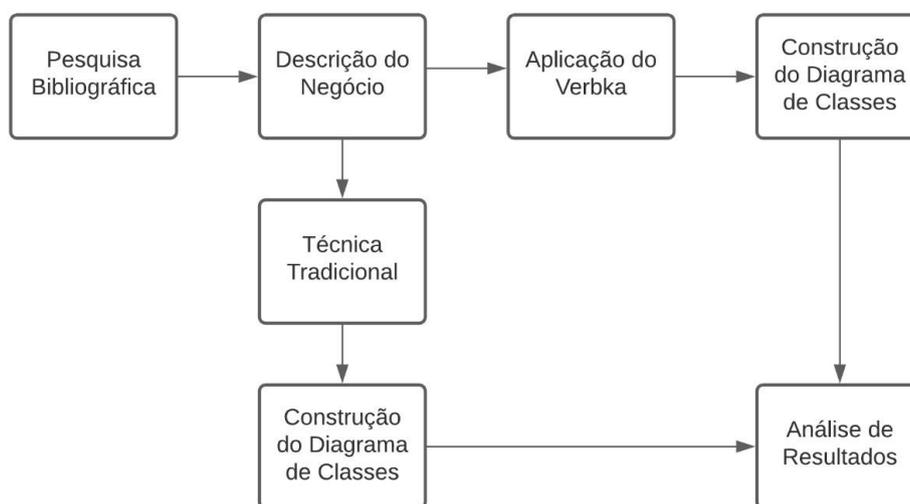


Figura 1: Sequenciamento das atividades do estudo

Fonte: Elaborado pelos autores

Dessa maneira, para a construção dos diagramas de classe utilizou-se a ferramenta Astah¹. Uma análise comparativa foi elaborada entre os dois diagrama, com objetivo de

¹ <https://astah.net/>

identificar as semelhanças entre os diagramas e informações faltantes que pudesse existir em um deles.

2 Fundamentação Teórica

2.1 Engenharia de Requisitos

A engenharia de requisitos representa um processo de compreensão e definição dos serviços requisitados do sistema e identificação de restrições relativas à operação e ao desenvolvimento do sistema (SOMMERVILLE, 2019). O autor citado anteriormente nos informa ainda que, para um engenheiro de software, uma das tarefas mais difíceis e importantes a ser enfrentada pelo profissional, é o processo de entendimento dos requisitos.

O processo da engenharia de requisitos tem como objetivo produzir um documento de requisitos acordados que especifica um sistema que satisfaz os requisitos dos *stakeholders* (SOMMERVILLE, 2019). Ainda segundo Sommerville (2019), existem quatro atividades principais do processo de engenharia de requisitos:

- Estudo de viabilidade: é feita uma estimativa acerca da possibilidade de se satisfazerem as necessidades do usuário identificado utilizando tecnologias atuais de software e hardware.
- Elicitação e análise de requisitos: esse é o processo de derivação dos requisitos do sistema por meio da observação dos sistemas existentes, além de discussões com os potenciais usuários e compradores, análise de tarefas, entre outras etapas.
- Especificação de requisitos: é a atividade de traduzir as informações obtidas durante a atividade de análise em um documento que defina um conjunto de requisitos.
- A validação de requisitos: essa atividade verifica os requisitos quanto a realismo, consistência e completude. Durante o processo, os erros no documento de requisitos são inevitavelmente descobertos.

2.1.1 Levantamento de Requisitos

Para Sommerville (2019) a descoberta dos requisitos, também chamada de levantamento de requisitos, é um processo de reunião das informações sobre o sistema

requerido e os sistemas existentes, separando as informações levantadas, os requisitos de usuário e de sistema. As fontes de informação durante a fase de descoberta de requisitos incluem documentação, *stakeholders* do sistema e especificações de sistemas similares. Assim algumas das principais técnicas de levantamento de requisitos segundo (SOMMERVILLE, 2019) são:

Entrevistas: Na maioria dos processos de engenharia de requisitos, são realizadas entrevistas formais ou informais com os *stakeholders* do sistema. Nestas entrevistas, a equipe de engenharia de requisitos questiona os *stakeholders* sobre o sistema que usam no momento e sobre o Sistema que será desenvolvido. Com isso, requisitos surgem a partir das respostas a tais perguntas.

Cenários: Os cenários também são conhecidos como histórias de usuários e podem ser particularmente úteis para adicionar detalhes a uma descrição geral de requisitos. Trata-se de descrições de exemplos de sessões de interação. Cada cenário geralmente cobre um pequeno número de interações possíveis. Diferentes cenários são desenvolvidos e oferecem diversos tipos de informação em variados níveis de detalhamento sobre o sistema.

Etnografia: Etnografia é uma técnica de observação que pode ser usada para compreender os processos operacionais e ajudar a extrair os requisitos de apoio para esses processos. Um analista faz uma imersão no ambiente de trabalho em que o sistema será utilizado. O trabalho do dia a dia é observado e são feitas anotações sobre as tarefas reais em que os participantes estão envolvidos. O valor da etnografia é ajudar a descobrir requisitos implícitos do sistema aos quais refletem as formas reais com que as pessoas trabalham, em vez de refletir processos formais definidos pela organização.

Diagrama de Classe: Os diagramas de classe são utilizados no desenvolvimento de um modelo de sistema orientado a objetos para mostrar as classes de um sistema e as associações entre estas classes (BOOCH; RUMBAUGH; JACOBSON, 2012). Uma classe de objeto pode ser pensada como uma definição geral de um tipo de objeto do sistema. Quando se está desenvolvendo modelos, durante os estágios iniciais do processo de engenharia de software, os objetos representam algo no mundo real, como um paciente, uma receita médica, um médico, entre outros (SOMMERVILLE, 2019). No desenvolvimento de um modelo, o primeiro estágio geralmente é o de olhar para o mundo, identificar os objetos essenciais e representa-los como classes (SOMMERVILLE, 2019). Para o Sommerville (2019), a maneira mais simples de realizar essa tarefa é escrever o nome da classe em uma caixa, além de simplesmente observar a

existência de uma associação, traçando uma linha entre as classes. Na seção 5 apresentaremos um exemplo de diagrama de classe contendo todas as ações, entidades e relacionamentos fornecidos pelo processo Verbka.

2.2 Trabalhos Relacionados

A pesquisa buscou estudos cujo foco eram extração de conhecimento aplicados no desenvolvimento de software especificamente na fase de análise de requisitos. Essa pesquisa bibliográfica evidenciou os trabalhos de Nasiria (2020), Elallaoui (2018), Narawita (2017) e Vasques *et al.* (2019).

Nasiria (2020) em seu estudo buscou estabelecer regras para melhorar a extração de conhecimento através das histórias de usuários, buscando aplicar essas informações juntamente aos métodos ágeis para assim extrair os elementos necessários a montagem do diagrama de classes. Para alcançar esses objetivos foi utilizado a linguagem *Natural Language Processing* - PLN (Processamento de Linguagem Natural) como modo de automação pretendida, tornando a transição das informações geradas pelas histórias de usuários em dependências e relacionamentos. Em seus resultados foi gerado uma taxa de precisão significativa de 98% tornando estes resultados dado como satisfatório.

Elallaoui (2018) estudou o efeito da análise da linguagem PLN aplicado a metodologia ágil Scrum² utilizando como base as histórias de usuário. Sua pesquisa buscou transformar as informações dos usuários em dados de entrada para o diagrama de casos de uso. As informações necessárias para criar o diagrama de casos de uso estão contidas nas histórias de usuários as quais poderão ser extraídas utilizando as técnicas em análise. Elallaoui (2018) mostrou que a utilização da extração de informações baseadas em histórias de usuários podem facilitar e simplificar a interpretação das informações fornecidas durante a captação das histórias de usuário. Como resultado, (ELALLAOU, 2018) apontou na investigação que com a análise da PLN, é possível reduzir ambiguidades, melhorar a compreensão dos dados, tornando mais simples a montagem do projeto, auxiliando os desenvolvedores.

Narawita (2017) utilizou inteligência artificial para que por meio da união de técnicas de PLN e Inteligência Artificial, fosse possível a criação automática de um diagrama de casos de uso. Sua abordagem consiste em analisar semanticamente as

² <https://www.scrum.org>

histórias de usuário através de algoritmos aumentando assim a precisão dos resultados. Segundo Narawita (2017), seus estudos apontaram que utilizando o algoritmo produzido somados às técnicas de PLN é possível a construção de diagramas nos mais diversos cenários tornando o processo preciso, eficiente e confiável.

Vasques *et al.* (2019) aplicaram o processo Verbka para apoiar a elaboração de diagramas de caso de uso na fase de pré-modelagem de software. Os autores citados anteriormente, utilizaram a descrição de negócio de um projeto de software disponível em um livro que abordava o assunto sobre a UML (VASQUES *et al.*, 2019). Com base na descrição de negócios, foi aplicado o Verbka gerando a tabela de proto-papéis, mapa conceitual e um diagrama de caso de uso de alto nível. Segundo Vasques *et al.* (2019) os principais resultados mostraram que com a aplicação do processo é possível obter uma lista contendo todos os requisitos do projeto, permitindo a elaboração dos diagramas de caso de uso, já que nessa lista é capaz de identificar os atores e seus relacionamentos.

Os estudos mencionados buscaram extrair informações relevantes ao processo de pré-modelagem de software, baseando sempre em histórias de usuários. Os trabalhos citados procuram alimentar um repositório de informações para auxiliar a construção de diagrama de casos de uso e diagrama de classes.

No trabalho de Vasques *et al.* (2019) os autores aplicaram o processo Verbka em uma descrição de negócio de um livro e focaram na elaboração do diagrama de caso de uso. Neste artigo o Verbka foi aplicado em uma descrição de negócio de uma empresa real e o foco foi na construção do diagrama de classe em nível conceitual. Em ambos os trabalhos o Verbka foi aplicado na fase de pré-modelagem. Assim, o objetivo deste artigo foi obter dados, analisar e organizar usando o Verbka para simplificar a construção do diagrama de classes em nível conceitual.

3 Processo Verbka

De acordo com Vasques *et al.* (2016), Verbka é um processo de aquisição de conhecimento com foco na semântica verbal, aplicado na extração de informação a partir de textos. O processo possui sete etapas (VASQUES *et al.*, 2016). Porém, para essa aplicação utilizamos apenas as etapas descritas a seguir (VASQUES *et al.*, 2016):

1. **Seleção Verbal:** Nesta etapa todos os verbos do texto devem ser identificados e marcados.
2. **Conversão para voz ativa:** A partir da identificação do verbo na voz

passiva no texto, é necessário fazer a conversão dos verbos para voz ativa.

3. **Geração da tabela de Proto-papéis:** Após a seleção e a conversão verbal, é necessário elaborar a tabela de proto-papéis, que consiste na fragmentação das sentenças identificando atores, ações, permissões e sujeitos aos quais as ações e/ou permissões serão aplicadas.

4. **Geração da tabela de Proto-papéis estendida:** Após a fragmentação das sentenças, gera a tabela de proto-papéis contendo as principais informações referente as sentenças extraída do texto, sendo necessário fragmentar novamente as proposições para deixar mais completas as informações extraídas. Assim, é necessário responder a questões adicionais para o verbo, sendo: “Para quem?”, “Onde?”, “Quando?”.

4 Aplicação do Verbka na Pré-Modelagem de Software

Tomamos como base para a aplicação do processo Verbka uma descrição de negócio real de uma empresa de desenvolvimento de software do município de Andradina-SP. Por motivos de segurança foram omitidas quaisquer informações que permitam identificar a empresa e/ou pessoas envolvidas. O foco do estudo foi na tela de venda de produtos alimentícios. O processo de venda de produtos envolve desde o cadastro do cliente ao gerenciamento do pagamento, permissões de acesso e outros módulos. A Tabela 1 mostra a lista de proto-papéis estendida após a aplicação do processo Verbka.

Tabela 1: Proto-papéis estendida

Nº	Quem?	Verbo	O que?/Quem?	Para quem?	Onde?	Quando?
1	Usuário	Consultar	o cliente, caso já seja cadastrado	Cliente	Tela Balcão	Durante a Venda
2	Sistema	Permitir	até quatro tipos de sabores para uma mesma pizza	Cliente	Tela Balcão	Durante a Venda
3	Sistema	Permitir	vincular balanças ao sistema ou de forma manual de acordo com a necessidade do cliente	Cliente	Sistema	-
4	Sistema	Possuir	controle de acesso a diferentes níveis, sendo totalmente customizável para proporcionar a melhor experiência para o usuário	Usuário	Sistema	-
5	Sistema	Atribuir	permissões de forma independente, seja para um formulário ou relatório	Usuário	Sistema	-
6	Sistema	Incluir	Toda a descrição necessária no cadastro de produtos	Usuário	Cadastro de Produto	Cadastro de produto

7	Sistema	Atribuir	o produto a grupos específicos previamente cadastrados	Usuário	Cadastro de Produto	Cadastro De Produto
8	Sistema	Permitir	selecionar um conjunto de produtos que possuem maior demanda	Usuário	Tela Balcão	Durante a venda
9	Usuário	Poder	aumentar ou diminuir a quantidade de cada item ou opções adicionais, como molhos ou outras características	Usuário	Tela Balcão	Durante a Venda
10	Usuário	Incluir	observações no pedido	Cliente	Tela Balcão	Durante a Venda
11	Usuário	Marcar	opção de produto(s) para viagem	Cliente	Tela Balcão	Durante a Venda
12	Sistema	Pesquisar	por produtos e/ou clientes funciona de modo universal, buscando os termos digitados em todas as categorias, ou pode-se utilizar filtros específicos	Usuário	Tela Balcão	Durante a Venda
13	Sistema	Solicitar	Acesso administrativo, conforme regras do negócio	Usuário	Operações em Geral	Execução
14	Usuário	Poder	acessar a tela de pagamento	Usuário	Tela Balcão	Durante a Venda
15	Sistema	Possuir	opções flexíveis de pagamento	Cliente	Tela pagamento	Durante o pagamento
16	Usuário	Poder	usar as formas de pagamento de modo combinado: cartão de crédito ou débito, dinheiro, cheque, boleto ou venda a prazo	Cliente	Tela pagamento	Durante o pagamento
17	Usuário	Selecionar	o desconto previamente cadastrado no Sistema	Cliente	Tela pagamento	Durante o pagamento
18	Usuário	Selecionar	um valor de desconto acima do permitido é necessário permissão de um supervisor e/ou possuir nível de acesso com função liberada			Durante o pagamento
19	Sistema	Gerar	troco	Cliente	Tela pagamento	Durante o pagamento
20	Sistema	Fornecer	o controle de vendas a prazo, onde fica registrado no sistema a identificação do cliente, valor, além da impressão de vias para coletar a assinatura e posterior armazenamento	Cliente	Tela pagamento	Durante o pagamento
21	Sistema	Permitir	Encaminhar os pedidos para impressoras diferentes: vias para cozinha e bar, contendo número do pedido e itens do mesmo	Usuário	Tela Balcão	Durante a venda
22	Sistema	Imprimir	O documento fiscal do cliente e a via do estabelecimento após a impressão dos pedidos	Usuário	Tela Balcão	Durante a venda
23	Sistema	Gerar	NFC-e ou SAT são restritas conforme estado	Cliente	Tela pagamento	Durante o pagamento
24	Sistema	Validar	Seus dados de cadastro e os documentos são emitidos conforme legislação respeitando o estado de origem	Cliente	Tela pagamento	Durante o pagamento

Os dados contidos na Tabela 1, representam a extração de proposições a partir da aplicação do processo Verbka. Uma proposição representa a união entre dois ou mais conceitos conectados por palavras de ligação formando uma afirmação (NOVAK;

CANÃS, 2010). A partir dessa classificação é possível facilitar a criação de um mapa conceitual (VASQUES *et al.*, 2016). Para Vasques *et al.* (2016) a construção de um mapa conceitual pode ser classificada como um composição de conceitos que estruturam um conhecimento interligados por frases de ligação o qual originam proposições baseadas em eventos e ações. Dessa maneira, usando a tabela de proto-papéis (Tabela 1) e a concepção para criação de mapa conceitual de acordo com o processo Verbka, foi elaborado o diagrama de classes apresentado na Figura 2.

Para a construção do diagrama de classe, o processo de modelagem se inicia a partir da primeira linha da tabela de proposições, e segue na ordem de cada um dos elementos que a constitui. Esse processo se repete até que todas as proposições presentes na tabela sejam modeladas. Na Figura 2, cada conceito é identificado como uma classe. Os verbos são representados por setas que indicam a ligação entre as classes. Essa relação entre os conceitos é aquela fornecida pelas preposições (com, em, para, etc.).

Para a criação do diagrama de classes da Figura 2, foi usado como base a Tabela 3, onde a coluna "Quem?" representa o responsável pela ação. A coluna "Verbo" representa a ação ou permissão definido na coluna "Quem?". Os complementos "O que?/Quem?" identifica seus métodos ou permissões de acesso a determinada ação do sistema. As classes são denominadas a partir da coluna "O que?/Quem?", em que é possível extrair qual componente do sistema é chamado para atender a usuários, como exemplo a balança, a execução do pagamento ou ainda o gerenciamento de descontos.

Analisando as colunas "Para quem?, Onde? e Quando?" (Tabela 1), é possível identificar relações adicionais como: relação bilateral entre usuário e sistema, cliente e pagamento e cliente e o sistema de frente de caixa (tela balcão). Por meio destas colunas é possível perceber e analisar a relação indireta entre as classes do Sistema, como por exemplo: "a interação do cliente na tela frente de caixa (balcão) durante o processo de venda de um produto".

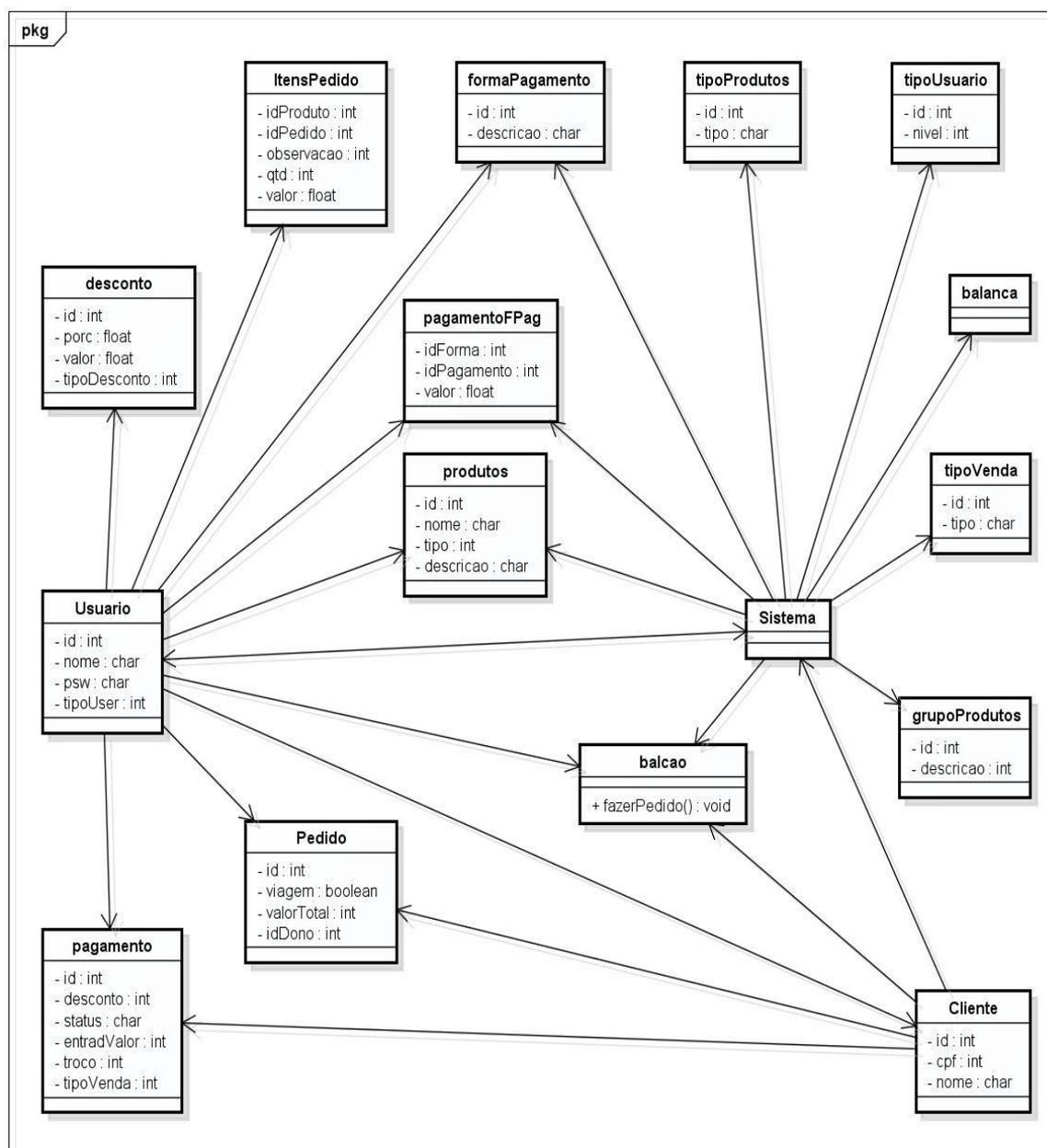


Figura 2: Diagrama de Classe nível conceitual - Processo Verbka

Fonte: Elaborado pelos autores

A Figura 3 a seguir, representa um diagrama de classes elaborado a partir das informações coletadas diretamente da descrição do sistema, sem a aplicação do processo Verbka.

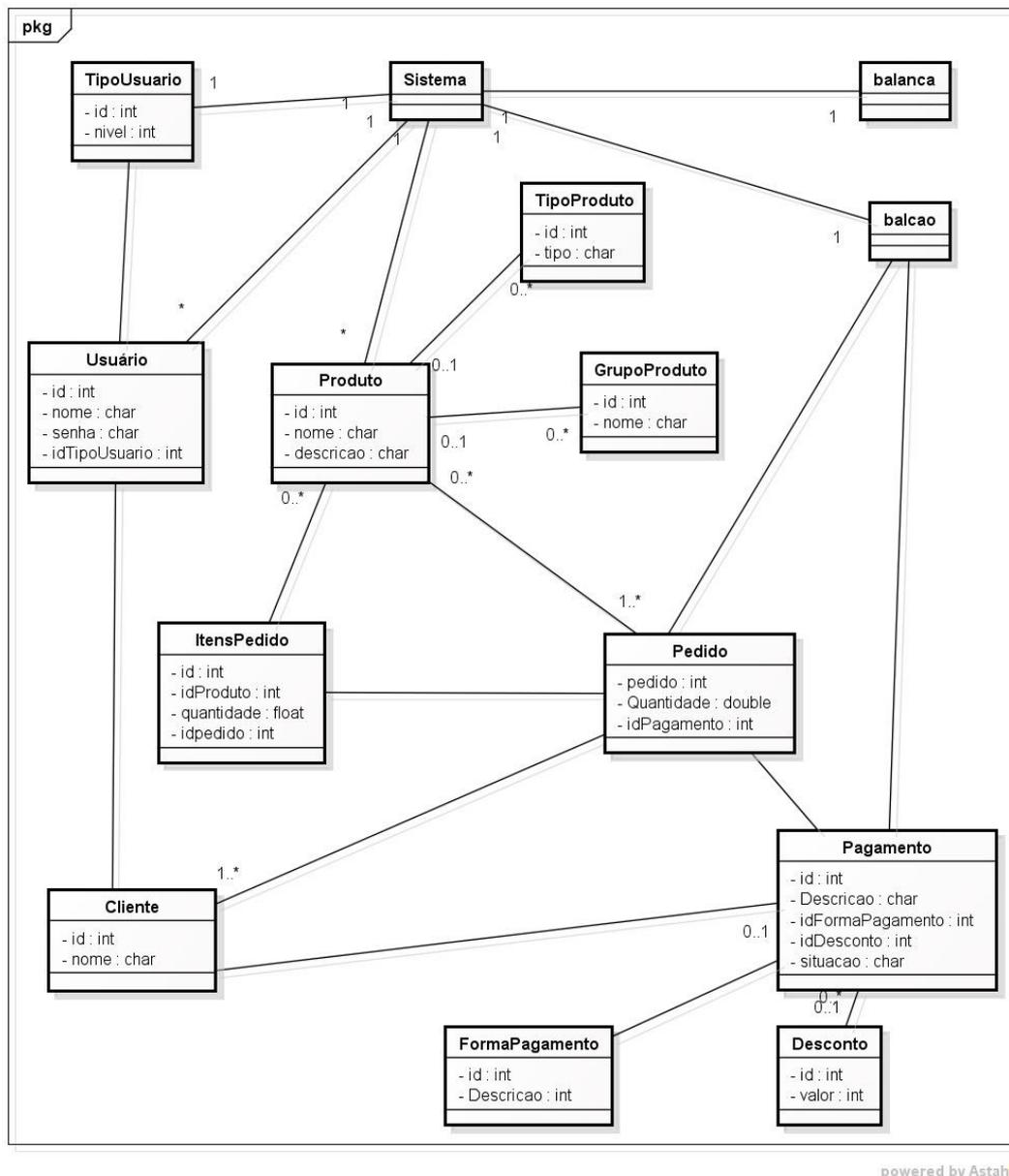


Figura 3: Diagrama de Classe nível conceitual - Forma Tradicional

Fonte: Elaborado pelos autores

O diagrama apresentado na Figura 3, foi elaborado a partir da descrição geral de um sistema, sem nenhuma técnica modelo ou resultado, utilizando apenas a forma tradicional de elaboração. Na Figura 2, que contém o diagrama criado baseando-se no processo Verbka, temos a adição de uma classe para classificar as vendas (tipoVenda) e a forma de pagamento (formapagamento). Inicialmente, as classes adicionais não foram detectadas utilizando a forma tradicional, como estamos no processo de pré-modelagem é possível que nas etapas subsequentes elas sejam apontadas.

5 Análise de Resultados

A experiência dos autores com a elaboração do primeiro diagrama usando o Verbka pode ter afetado a elaboração do segundo diagrama que compromete a validade interna, já que as aplicações foram realizadas somente pelos autores. Assim sendo, para validação dos dados, seria necessário mais aplicações com diferentes pessoas envolvidas. Creswell e Clark (2015) aponta que é necessário o embasamento em mais de um caso e assim obter um padrão de comportamento, afim de identificar resultados válidos.

O diagrama elaborado a partir do processo Verbka foi possível identificar as classes: desconto, usuário, pagamento, itensPedido, pedido, pagamentoFPag, produtos, formaPagamento, balcão, sistema, tipoProduto, tipoUsuario, balança, tipoVenda, grupoProdutos e clientes totalizando 16 (dezesseis) classes. Já no segundo diagrama de classes sem o uso do Verbka foi identificado as seguintes classes: tipoUsuario, usuario, cliente, sistema, produto, itensPedido, tipoProduto, grupoProduto, pedido, balança, balcão, pagamento, desconto e formaPagamento totalizando 14 (quatorze) classes.

Assim sendo, durante a execução do Verbka não foi necessário consultas constantes na descrição geral do sistema, pois todas as informações estavam na Tabela de Proto-papeis estendida (Tabela 3). Já na elaboração do segundo diagrama de classes utilizando a forma tradicional, foi necessário constantemente consultar a descrição geral do sistema para confirmação das classes identificadas e entendimento do sistema.

O uso da Tabela de Proto-papeis estendida facilitou o entendimento do assunto, a familiarização dos dados e a elaboração do diagrama de classes. Assim, o processo de extração das classes, contidas na coluna dos complementos fica clara, sem necessidade de compreender com detalhes os requisitos do sistema para a criação das classes. Já durante a elaboração do diagrama de classe na forma tradicional, foi necessário ter mais atenção as regras de negócio para a identificação de possíveis classes e atributos.

Observa-se que na Figura 2 o processo Verbka identificou "tipoVenda" como uma classe, porém esta classe poderia ser apenas um atributo da classe "Pagamento". De modo igual com a classe "pagamentoFPag" (Figura 2) que apresenta atributos que poderiam em uma análise e de acordo com as restrições do projeto fazer parte da classe "formaPagamento", eliminando assim essa classe.

O uso do processo Verbka aplicado na fase de pré-modelagem colaborou com a identificação dos principais conceitos (classes) da descrição geral do sistema, o que pode contribuir com o entendimento do sistema e seus processos de negócio. Os verbos de

ações identificados na tabela de proto-papéis podem representar no diagrama de classes os métodos da classe. No entanto, nessa extração como o objetivo foi elaborar um diagrama de classe que atendesse ao nível conceitual, essa opção não foi considerada.

Outro ponto importante em relação ao diagrama de classe elaborado a partir da tabela proto-papéis é que a maioria das classes não estão conectadas entre si, mas conectadas com a classe principal (conceito raiz). Como o modelo tem o foco na pré-modelagem, após sua elaboração os *stakeholders* responsáveis pela modelagem do software, devem iniciar o processo de inferências. O objetivo do processo de inferências é conectar novas classes (conceitos) gerando assim novos relacionamentos e extraíndo novos conhecimentos, que pode representar no modelo novos processos de negócio.

Considerações Finais

A aplicação do processo Verbka na pré-modelagem de software facilitou a compreensão e identificação das classes. O processo Verbka permitiu que os conceitos fossem separados em formato de tabela, o que ficou mais claro a diferenciação entre classes e atributos.

A execução deste estudo apontou que, com a utilização do processo Verbka durante a etapa de pré-modelagem foi possível favorecer a extração de dados, uma vez que subdividimos a aplicação do método em etapas. Este artigo abordou a modelagem conceitual na fase de pré-modelagem com uso do diagrama de classes. As demais etapas de desenvolvimento de software não foram objeto de estudo neste artigo.

Como trabalhos futuros pretende-se realizar novas aplicações do processo Verbka para construção do diagrama de classe, considerando outras etapas e classificações do Verbka, como por exemplo o uso das setas do processo (VASQUES *et al.*, 2016), que poderiam ser usada para especificar as ligações entre as classes em relação aos tipos de relacionamentos existentes na construção do diagrama de classe.

Referências

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: guia do usuário. 2. ed. Rio de Janeiro: Elsevier, 2012.

CRESWELL, John W.; CLARK, Vicki L. **Plano. Pesquisa de Métodos Mistos**. Porto Alegre: Penso Editora, 2015. (Série Métodos de Pesquisa).

ELALLAOUI, Meryem; NAFIL, Khalid; TOUAHNI, Raja. Automatic transformation of user stories into UML use case diagrams using NLP techniques. **Procedia computer science**, v. 130, p. 42-49, 2018.

GODOY, Arilda Schmidt. Pesquisa qualitativa: tipos fundamentais. **Revista de Administração de Empresas**, v. 35, n. 3, p. 20-29, 1995.

NARAWITA, Chamitha Ramal *et al.* UML Generator-Use Case and Class Diagram Generation from Text Requirements. **ICTer**, v. 10, n. 1, p. 1-10, 2017.

NASIRI, Samia *et al.* Towards a Generation of Class Diagram from User Stories in Agile Methods. **Procedia Computer Science**, v. 170, p. 831-837, 2020.

NOVAK, Joseph D.; CAÑAS, Alberto J. A teoria subjacente aos mapas conceituais e como elaborá-los e usá-los. **Práxis educativa**, v. 5, n. 1, p. 9-29, 2010.

SOMMERVILLE, I. **Engenharia de software**. 10. ed. São Paulo: Person Universidades, 2019.

VASQUES, Dildre G. *et al.* An approach to knowledge acquisition based on verbal semantics. *In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES (HICSS)*, 49., **IEEE**, 2016. p. 4144-4153.

VASQUES, Dildre G. *et al.* Use Case Extraction through Knowledge Acquisition. *In: ANNUAL INFORMATION TECHNOLOGY, ELECTRONICS AND MOBILE COMMUNICATION CONFERENCE (IEMCON)*, 10., **IEEE**, 2019. p. 0624-0631.